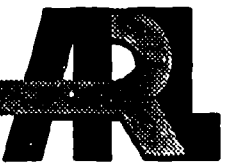
ARMY RESEARCH LABORATORY



Evaluation of a Multiple Instruction/ Multiple Data (MIMD) Parallel Computer for CFD Applications

Stephen J. Schraml

ARL-TR-589

October 1994

NOTICES

Destroy this report when it is no longer needed. DO NOT return it to the originator.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1994	Final, April 1991 - November 1992

4. TITLE AND SUBTITLE
Evaluation of a Multiple Instruction/Multiple Data (MIMD) Parallel Computer for CFD Applications

5. FUNDING NUMBERS
WO: 4G592-421-U2-U042

6. AUTHOR(S)
Stephen J. Schraml

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-WT-NC
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-OP-AP-L
Aberdeen Proving Ground, MD 21005-5066

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
ARL-TR-589

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

In an attempt to evaluate the merits of massively parallel processing computers for the numerical simulation of blast phenomena, the U.S. Army Research Laboratory (ARL) has adapted one of its blast modeling tools to several unique parallel architectures. This report describes the adaptation of the BRL-Q1D code, a quasi-one-dimensional, finite difference Euler solver, to the Intel iPSC/860 parallel supercomputer. The code was reconfigured for the iPSC/860 using FORTRAN 77 and the Intel iPSC/860 message-passing library. The performance of the code was measured on the iPSC/860 for a variety of problem sizes and processor configurations. The performance was found to be highly dependent on the size of the problem. This problem size dependency was most noticeable when fewer processors were employed. Results of scalability tests indicate that the code performance scales in a roughly linear fashion about extrapolated lines of ideal performance.

14. SUBJECT TERMS	15. NUMBER OF PAGES
nuclear explosion simulation, shock tubes, computer programming	26
	16. PRICE CODE

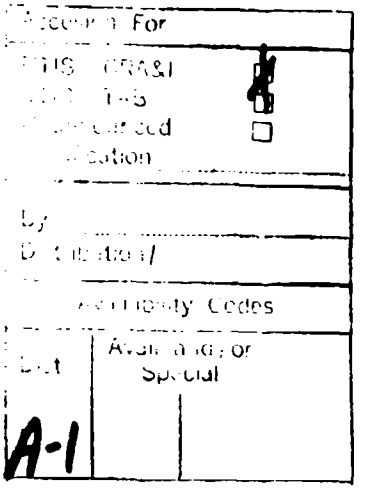
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Intentionally Left Blank

Acknowledgments

The BRL-Q1D code adaptation and performance trials were conducted with the resources provided by the U.S. Army Research Laboratory (ARL), the State University of New York at Stony Brook and the California Institute of Technology. The author acknowledges the assistance provided by the support staffs of these organizations.

The author also acknowledges Mr. Monte W. Coleman, ARL, for assistance provided in executing the calculations described in this report, and Mr. Thomas M. Kendall, also of ARL, for reviewing this report.



Intentionally Left Blank

Table of Contents

Intentionally Left Blank

List of Figures

Intentionally Left Blank

1. Background

Due to the rising costs of large-scale experimentation and the uncertainty of scaling effects in experiments, the U.S. Army is becoming increasingly dependent on computer simulations to assess the vulnerability of military systems to nuclear blast. Present vector supercomputer technology can provide detailed fluid dynamic simulations in two dimensions in a production environment (less than 10 CPU hours). Three-dimensional simulations employing limited spatial resolution, which are only sufficient for modeling relatively simple geometries, still require 100 or more CPU hours on a vector supercomputer.

Obviously, these low resolution simulations with simple geometries do not provide sufficient information about the vulnerability of specific military systems to the overturning or crushing effects of blast produced by tactical nuclear weapons. To provide accurate assessments of system vulnerability, highly detailed three-dimensional simulations with coupled fluid-structure interaction are required. To make this type of numerical simulation possible, increases in supercomputer performance of two or more orders of magnitude must be realized.

The rapidly maturing field of massively parallel processing (MPP) has the potential to offer the compute performance required for detailed three-dimensional fluid dynamic simulations. There are many different types MPP machines available on the computer market today. However, generally speaking, most current MPP computers have the following two basic characteristics:

1. They combine the resources of a large number of processors to simultaneously solve different parts of a large problem.
2. Each processor has its own bank of local memory.

Particular MPP computers differ primarily in the way the processors access data in their own memory and data in the memory of other processors. These data access methods typically define the programming methods which are required to extract maximum performance from all of the resources that the machine has to offer.

As a means of evaluating MPP technology, the U.S. Army Research Laboratory (ARL) continuously adapts one of its blast modeling tools to emerging MPP computer platforms.[1] Through continuous evaluation of MPP computers, the ARL can configure its software tools to exploit this technology, thus making detailed three-dimensional fluid dynamic simulations available in a production computing environment.

2. The iPSC/860

The Intel iPSC/860 was the parallel computer chosen for the adaptation and evaluation described in this report. The iPSC/860 is a Multiple Instruction / Multiple Data (MIMD) parallel computer. This implies that the processors of the computer can perform a number of

different operations simultaneously, if requested to do so. This is quite different from a Single Instruction / Multiple Data (SIMD) computer, in which all of the computers processors perform identical operations, simultaneously, on different sections of data.

The heart of the iPSC/860 is a set of Intel i860 microprocessors. Each of these processors has a fixed amount of local memory. Systems configured this way are often referred to as "distributed memory multiprocessors." Most current MPP machines are distributed memory architectures. The individual processors of the iPSC/860 are linked together by an interconnect network which allows high bandwidth data transfer between processors. The earlier model of the iPSC/860, known as the Gamma, employs a hypercube topology as the interconnect network.[2] A hypercube can be envisioned as a large number of nested cubes with each point of a cube representing a node, or processor, in the system. The later model iPSC/860, called the Delta, employs a two-dimensional mesh topology as the processor interconnect network. Each type of interconnect network is designed to have a maximum number of connections between available processors, while at the same time minimizing the distance that data must travel when moving from its originating processor to its destination processor.

Like most MPP machines, the iPSC/860 is designed to be a scalable system. As such, it should be possible to linearly increase the compute performance of an application by increasing the number of processors allocated to the application. Thus scalability is the motivation in the development of MPP computers. In a truly scalable system, a desired level of performance can be obtained by simply acquiring the necessary quantity of processors. This is potentially more cost-effective than obtaining performance gains through advances in single processor design.

Unfortunately, building a scalable architecture is only half of the solution to obtaining increased performance. To optimally employ all of the resources provided by MPP computers, the application's algorithm must be designed to be scalable as well. Optimum algorithm design for the iPSC/860, and most other MIMD machines, is accomplished through a style of programming known as message-passing. When a parallel application is run on the iPSC/860, the data is distributed among the available memory of the processors being used. If a particular processor needs access to a piece of data stored in another processor's memory to perform a calculation, then that data must be transferred from the originating processor to the processor which needs the data. To accomplish this data transmission, the application must be written to explicitly pass the data from the original processor, to the target processor.[3]

One potential bottleneck in distributed memory parallel computers is this transfer of data between processors. Even though the processors are connected by a high-speed network, the time required to move data between processors is typically much greater than the time spent by the receiving processor executing a floating point instruction using that data. Consequently, the ultimate goal in developing an algorithm which employs message-passing techniques is to minimize time spent transferring data, thereby maximizing the time spent in computing the solution. With this in mind, the algorithm designer must be sure to transfer only that data which is necessary for the calculations to be performed correctly.

3. Blast Modeling Application

The BRL-Q1D code was selected as the blast modeling application which is used by the ARL to evaluate the programming environment and the computational performance of massively parallel computers. BRL-Q1D is a quasi-one-dimensional, finite difference, single material, polytropic gas fluid dynamics code and is primarily used to simulate flow in shock tubes. This code was chosen for its relative simplicity and its algorithmic similarities to the two- and three-dimensional codes that are currently used for blast modeling applications. Therefore, adaptation of this code is the most cost-effective means of evaluating massively parallel computer architectures. The similarities in the solution algorithms of BRL-Q1D and more complex multidimensional codes can provide insight to the potential performance of these more complex codes on MPP computers.

The BRL-Q1D code incorporates two computational techniques, an implicit finite difference technique [4] and an explicit finite difference scheme.[5] Only one of these algorithms may be used in a particular BRL-Q1D calculation. The solution scheme which is employed is determined by a set of user-defined input options. The solution algorithms are applied to the quasi-one-dimensional Euler equations in their weak conservative form.[6]

One multidimensional fluid dynamics code used extensively at ARL for the numerical simulation of blast effects is the SHARC code. SHARC is an explicit, finite difference Euler solver which is second-order accurate in space and time.[7] Because of its algorithmic similarities to the SHARC code, only the explicit algorithm of the BRL-Q1D code was adapted to the iPSC/860.

The MacCormack explicit scheme employed in BRL-Q1D is a second-order, non-centered, predictor-corrector technique that alternatively uses forward and backward differences for the two steps. The first step predicts the value of the state variables at a grid point based on the values of the grid point and its neighboring downstream grid point. The second step then corrects these state variables based on the values at the grid point and its neighboring upstream grid point.

Prior to its adaptation to the iPSC/860, the BRL-Q1D code existed in standard Fortran 77 form and the explicit, finite difference algorithm had been optimized for maximum performance on vector supercomputers. So that it could obtain maximum performance on the iPSC/860, the code was modified to evenly distribute the arrays among the available processors and then calls to the iPSC message-passing library were inserted where necessary for the transmission of data between processors.

The even distribution of arrays among available processors is a technique which is often referred to as "domain decomposition." In the case of the one-dimensional code, domain decomposition is nothing more than evenly dividing the number of available processors into the number of grid points being used by the calculation, then placing that number of adjacent grid points on successive processors. For example, to distribute an 800 grid point calculation among 8 processors, grid points 1 to 100 would be placed on processor 0, 101 to 200 on processor 1, etc. The BRL-Q1D code was modified in such a way that each time the code was run, it would automatically determine the number of processors that were available,

and dynamically allocate the arrays based on the grid size and this returned number of processors. Of course, domain decomposition in two or three spatial dimensions can be much more complex than this simple one-dimensional case. This is especially true when complex geometries are being modeled.

After the domain decomposition scheme had been developed, it was then imposed on the solution algorithm, so that the computational load would be evenly distributed among the processors. In the original Fortran 77 implementation of the BRL-Q1D code, the successive prediction and correction of state variables was accomplished through the use of **DO** loops which proceed through the one-dimensional grid, from beginning to end, in one grid point increments. For the message-passing implementation, the range of **DO** loop operation on a particular processor was limited to the grid points which were allocated to that processor. If a calculation required data from a grid point which does not reside on the processor doing the calculation, that data is passed to the processor prior to the calculation. The following examples of original Fortran 77 code and message-passing code illustrate this logic.

Segment of Original Fortran 77 Code

```
      do 20 j=2,jmax-1
        s(j,1) = q(j,1) - dt*(f(j+1,1)-f(j,1))
   20 continue
```

Segment of Equivalent Message Passing Code

```
      istart=ibeg(mynode()+1)
      istop =iend(mynode()+1)
      if (istart.eq.   1) istart=2
      if (istop .eq.jmax) istop =jmax-1
      if (numnodes().ne.1) call passleft(f,1)
      do 20 j=istart,istop
        s(j,1) = q(j,1) - dt*(f(j+1,1)-f(j,1))
   20 continue
```

These code segments are examples of a predictor step in the explicit algorithm. In these examples, the array s is being calculated from the arrays q and f and a constant, dt for all grid points between the second and the next to the last, inclusive. The calculations for the first and last grid points are performed in a separate boundary condition subroutine. The functionality of the Fortran 77 code is obvious from the example given above. In the message-passing example, the vectors *ibeg* and *iend* represent the beginning and ending array indices assigned to each processor in the domain decomposition subroutine. In the case of the first processor, the value of *ibeg* is reset from a value of 1 to 2 in order to conform to the limits on the **DO** loop in the Fortran 77 example. In a similar fashion, the value of *iend* is reset from *jmax* to *jmax-1*.

In the message-passing example, when the calculation of the state variable s reaches the last iteration in the loop on a particular processor, a value from the f array which resides on a neighboring processor is required. For this calculation to be performed properly,

all processors except the first pass the first element of *f* on the processor to the adjacent processor on the left. This operation is initiated by the call to the subroutine *passleft* in the example. The first argument in the call to the *passleft* subroutine is the name of the array to be passed. The second argument is the number of elements to be passed between processors. If the calculation of *s* had used a *f(j+2,1)* term, then all processors except the first would pass two elements of *f* to the left neighboring processor. For the the corrector step in the explicit algorithm, an analogous *passright* subroutine is used in which all processors except the last pass data from the specified array to the neighboring processor on the right.

To further illustrate the techniques employed in message-passing programming, the *passleft* subroutine is listed below. This subroutine listing shows the process by which processors pass the first *n* sub-array elements in their memory to their respective left neighboring processor. The receiving processors then store this data as the last *n* sub-array elements in memory.

In studying the function of this subroutine, it is important to remember that the subroutine runs simultaneously on all of the assigned processors. When the *passleft* subroutine is called, data from a particular flow parameter array are stored in a dummy array *a*. This dummy array, is a two-dimensional array with the first dimension assigned to be the number of grid points, *jmax*, and the second dimension assigned as the number of variables per grid point for the array, in this case three (energy, density and momentum, for example). The *passleft* subroutine is designed to transfer all three of these variables for a given grid point for any particular call to the subroutine. The listing of the *passleft* subroutine shows the following steps which are taken to transfer the data:

1. The send and receive indices for each processor are defined. These indices becomes counters in a **DO** loop if data from multiple grid points are to be transferred between processors.
2. All of the processors are synchronized in time so that the communication takes place simultaneously on the processors.
3. On all processors except the first, the three variables for a given grid point of the dummy array *a* are written into a temporary, three element array *b*. The particular grid point is defined by the send index.
4. All of the processors except the first send the contents of array *b* to the left neighboring processor.
5. All of the proccessors except the last receive the data sent from the right neighboring processor and store the data in the temporary, three element array *b*.
6. On all of the processors except the last, the three variables stored in the *b* array are written into the dummy array *a* for the proper grid point. The particular grid point is defined here by the receive index.
7. This process is repeated if data from more than one grid point is being transferred.

Listing of Passleft Subroutine

```
      subroutine passleft (a,n)
c
c     this subroutine passes the first n elements of sub-array
c     a from a processor to its left neighbor processor.
c     the receiving processor stores the data as the last n
c     elements of the sub-array a.
c
      include 'param.h'
      include 'mimd.h'
c
      dimension a(jmax,3),b(3)
c
c     is = send index for local node
c     ir = receive index for local node
c
      is = ibeg(mynode()+1)-1
      ir = iend(mynode()+1)
c
      do 10 k=1,n
        is=is+1
        ir=ir+1
        call gsync()
        if (mynode().ne.0) then
          do 20 j=1,3
            b(j) = a(is,j)
   20     continue
          call csend (0,b,3*4,mynode()-1,mypid())
        endif
        if (mynode().ne.numnodes()-1) then
          call crecv (0,b,3*4)
          do 30 j=1,3
            a(ir,j) = b(j)
   30     continue
        endif
   10 continue
c
      return
      end
```

4. Results

When the adaptation of the BRL-Q1D code was completed, the performance of the code was tested on both Gamma and Delta models of the iPSC/860 architecture. Due to the nature of the hypercube topology of the iPSC/860 Gamma model, the number of available processors is always an exact power of two. A Gamma model with 16 processors was employed for these tests. The iPSC/860 Delta model, with its mesh topology, is not constrained to the power of two processor requirement of the Gamma. The particular Delta machine used for the tests has 532 processors. However, only 256 processors were used in the maximum Delta configuration tests. So that Delta results could be directly compared with Gamma results, all Delta trials employed power of two processor configurations and problem sizes.

In all cases, the measured performance of the BRL-Q1D code was represented as the "whiz factor." This is a measure of the average CPU time required for the code to compute a solution, divided by product of the number of grid points and the number of cycles in the calculation ($\mu s/grid\ point/cycle$). This is a convenient method of measuring the code's performance because it normalizes the run time against the problem size and the number of time steps in the calculations. Thus using the whiz factor as a benchmark, results of different calculations can be compared directly. For a particular processor configuration and problem size, the reported performance is the minimum whiz factor (i.e., best performance) out of a set of several identical trials.

The first set of tests was performed to determine the influence of varying problem size on code performance. These tests were performed only on the Gamma model. The results of these tests are illustrated in Figure 1. This figure shows several curves illustrating the relationship between whiz factor and problem size for different processor configurations. This figure shows that, for small problems, the performance of the iPSC/860 is highly dependent on the size of the problem. As the problem size is increased, all of the processor configurations approach an upper limit on performance (i.e., a minimum whiz factor). All of the curves in Figure 1 have a similar shape; an initially sharp drop in whiz factor as the problem size is increased, followed by a bump in the middle of the curve, and ending with a leveling off as the maximum performance for that processor configuration is reached. The bump in the middle of each curve is a result of the increasing size of the problem filling the memory systems of each processor. The curve representing the trials with eight processors is slightly different from the other curves at the data points corresponding to problem sizes of 2^{10} and 2^{11}. For this processor configuration, the performance increased very little from a problem size of 2^9 to 2^{10}. Then, from 2^{10} to 2^{11} the performance increased significantly. In fact, the measured performance of the code on eight processors is exactly the same as the sixteen processor result for the same problem size of 2^{11}. Several additional trials were performed here to veryify the result, and results were consistent. Thus this appears to be merely an interesting characteristic of the Gamma model, most likely resulting from a fortuitously optimum layout of the data in memory for the particular configuration of eight processors running a problem size of 2^{11}.

As previously discussed, the Delta model allows the user to access arbitrary numbers of available processors. The Delta also allows users to define the two-dimensional layout of the processors within the mesh topology. With this in mind, a series of tests was performed to determine the influence of processor layout on code performance. In this series, a problem size of 16384 was run on 16 processors of the Delta. Five tests were performed in which processor layouts of 1x16, 2x8, 4x4, 8x2 and 16x1 were employed. The results of these tests are provided in Table 1. Also included in this table is the result of the identical calculation on the Gamma. These results illustrate that the performance of the BRL-Q1D code on the Delta is independent of processor configuration. Thus it can be assumed that communication of data between processors is not heavily influenced by the proximity of any two communicating processors.

Table 1. BRL-Q1D Performance on Delta as a Function of Processor Layout

Processor Layout	Whiz Factor ($\mu s/grid\ point/cycle$)
1x16	7.75
2x8	7.79
4x4	8.42
8x2	7.75
16x1	7.81
Gamma 2^4	9.00

As previously discussed, true scalability of both the architecture and the algorithm are essential to successful exploitation of MPP technology. Thus to determine the scalability of the BRL-Q1D code on the iPSC/860, a final series of tests was performed in which successive tests employed increasing numbers of processors. The problem size was accordingly increased with the increase in processors, so that lines of constant problem size per processor could be determined. These tests were performed on both the Gamma and the Delta models and are illustrated in Figures 2 through 6.

The results shown in Figures 2 to 6 illustrate the scalability tests of Gamma and Delta using 512 grid points per processor. In these figures, the measured performance data points for the Gamma are represented by the solid dots, while the data for the Delta is represented by the star symbols. If the adapted BRL-Q1D algorithm were perfectly scalable on the Gamma and Delta architectures, then doubling the number of processors used would result in a factor of two decrease in the whiz factor (i.e., doubling the number of processors would double the performance). The solid line in Figures 2 to 6 represents this ideal scalability which is extrapolated from the measured whiz factor for one processor of the Gamma. Accordingly, the dashed line represents the same ideal scalability relationship for the Delta.

Due to the logarithmic formulation of the scalability relationship, the ideal scalability curves result in straight lines when plotted against log-log axes. Figures 2 to 6 show that the lines of ideal scalabilty for both Gamma and Delta pass through the scatter of the measured data, indicating perfect scalability. These figures also illustrate that the performance of the Delta is slightly better than that of the Gamma for all cases. This improvement can be

attributed to advances in compiler technology and system software on the Delta, the later of the two architectures.

5. Conclusions

This report has outlined the successful implementation of the explicit, finite difference BRL-Q1D algorithm on the Intel iPSC/860 parallel computer. A division of labor among processors along with a coordinated use of message-passing between processors was used to evenly distribute the algorithm across the resources of the architecture. The results presented in the figures lead us to conclude that this message-passing implementation of the BRL-Q1D code is indeed perfectly scalable on both hypercube and mesh processor topology MIMD computers.

The likelihood of a successful implementation to parallel architectures is dependent on the level of inherent parallelism in the algorithm. The explicit BRL-Q1D algorithm is inherently data parallel. Its inner **DO** loops typically span the entire computational mesh. As a result, distribution of the algorithm across processors is easily accomplished.

As stated earlier, the adaptation of the BRL-Q1D code to the iPSC/860 was part of an attempt to evaluate MPP technology for blast modeling applications. The success of this and other implementations of the code on MPP computers is an indication that significant performance improvements can be obtained from the adaptation of large, multidimensional fluid dynamics codes to MPP platforms.

Other types of codes, however, may not be good candidates for adaptation to parallel computers. When this is the case, it may be necessary to completely restructure the basic algorithm in order to increase the level of inherent parallelism. Once this is done, then the algorithm can be adapted to parallel computers with greater likelihood of a successful implementation.

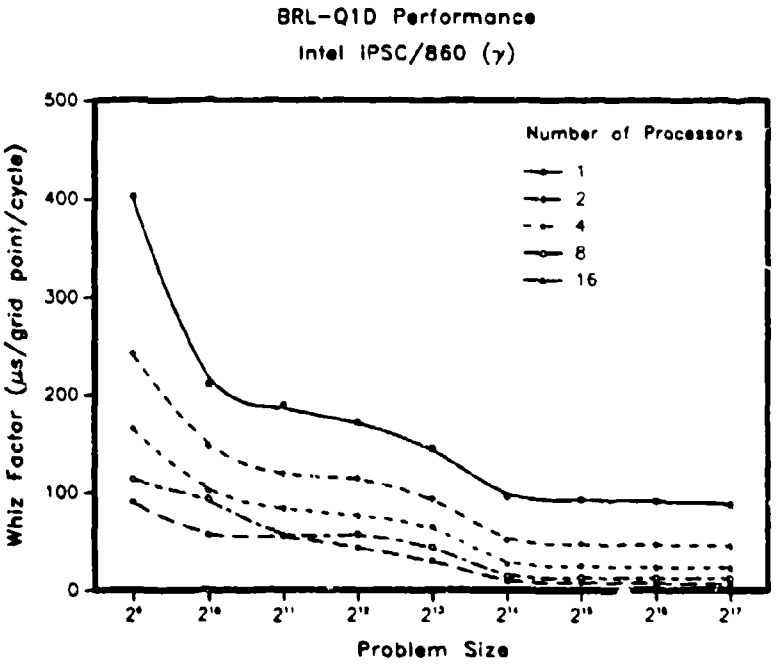


Figure 1. Measured Performance of BRL-Q1D Code

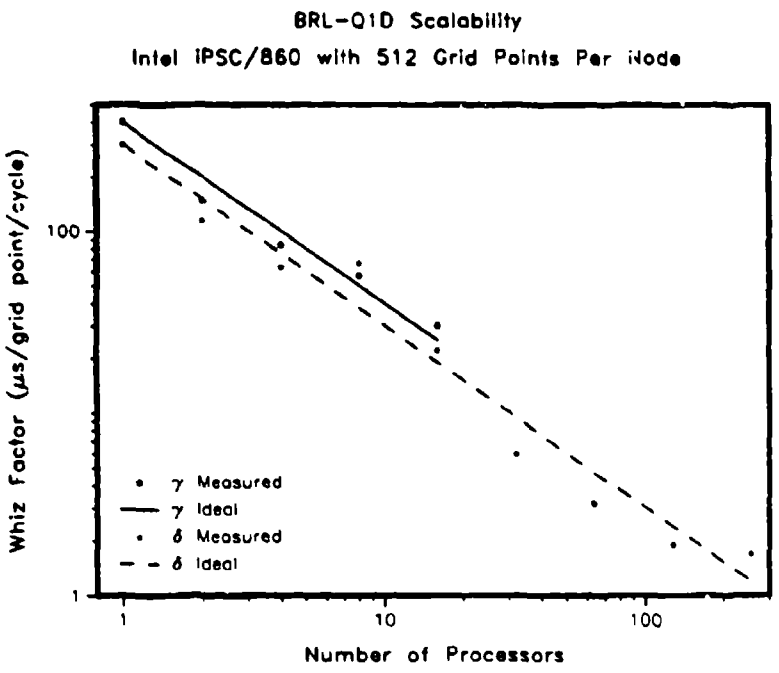


Figure 2. Scalability Using 512 Grid Points / Node

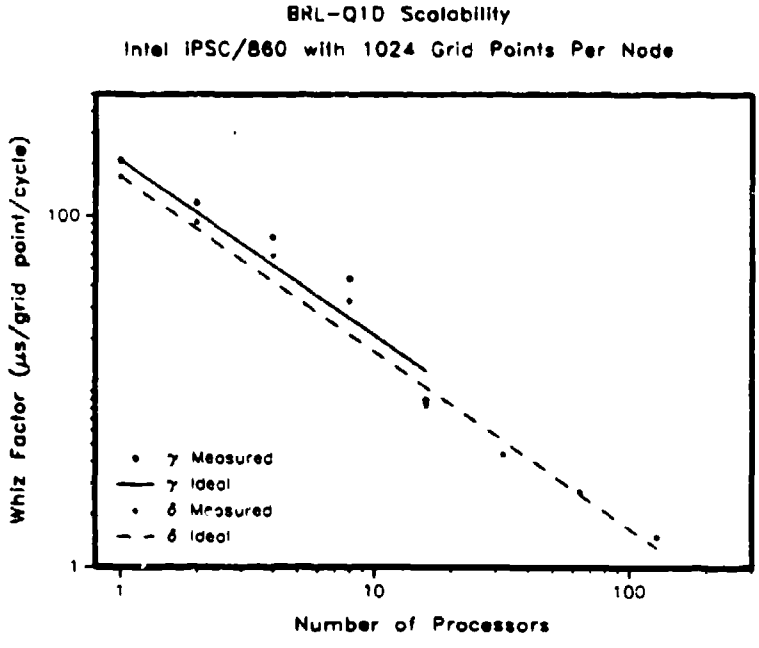


Figure 3. Scalability Using 1024 Grid Points / Node

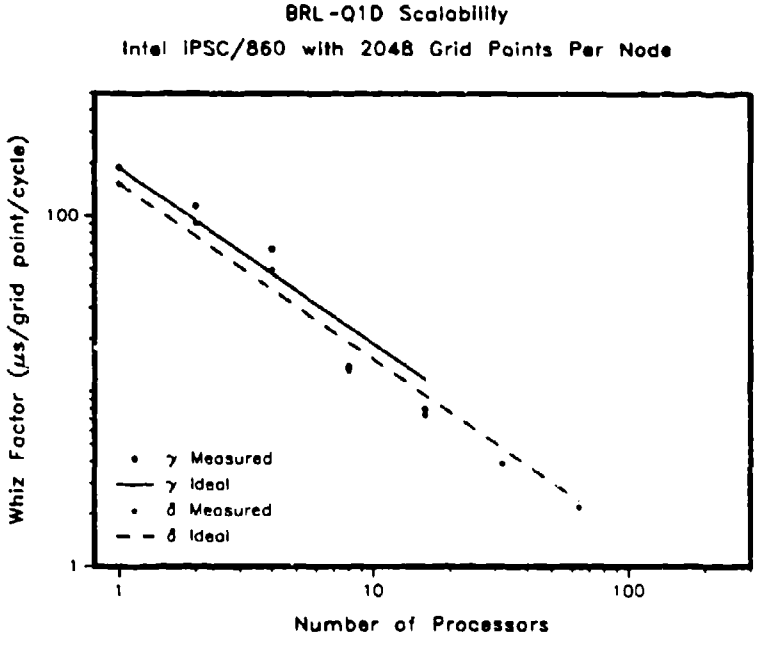


Figure 4. Scalability Using 2048 Grid Points / Node

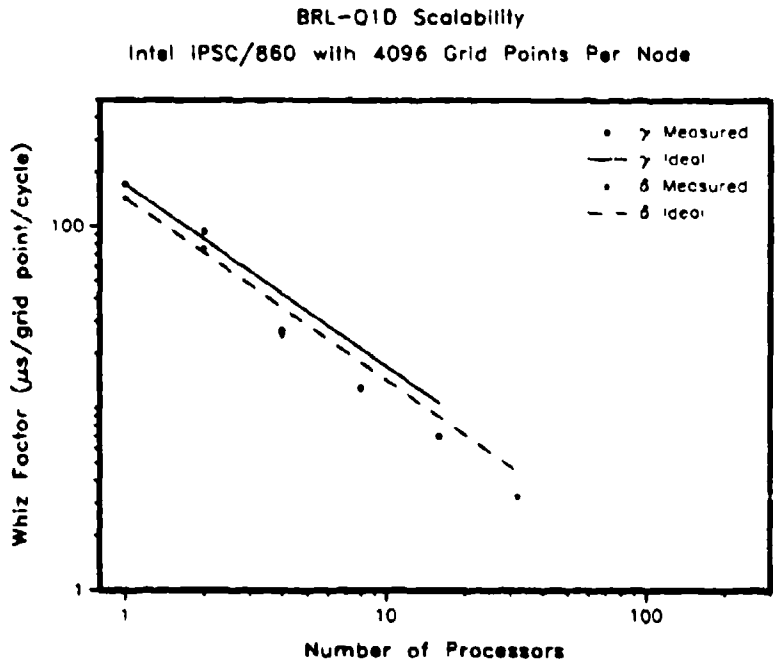


Figure 5. Scalability Using 4096 Grid Points / Node

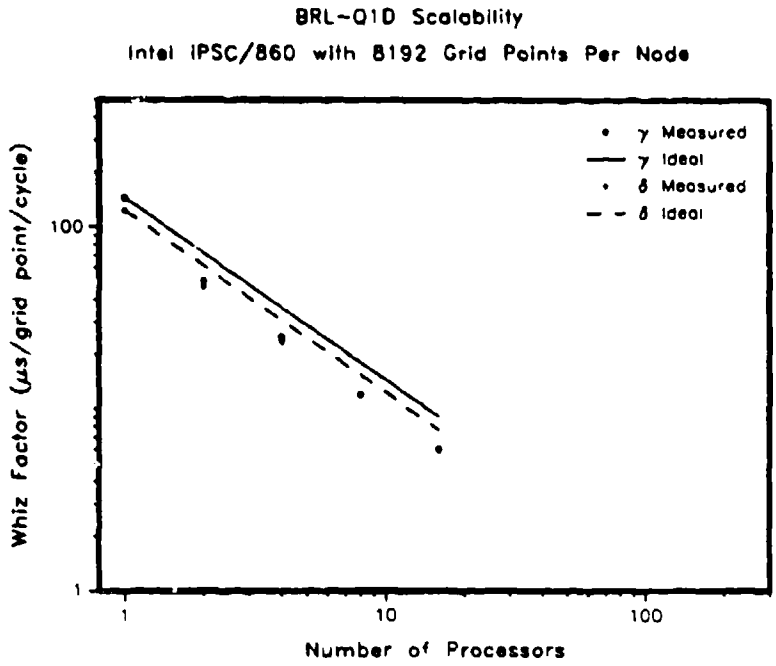


Figure 6. Scalability Using 8192 Grid Points / Node

References

1. Schraml, S.J. "A Data Parallel Implementation of the BRL-Q1D Code." Technical Report BRL-TR-3389. U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland. September 1992.

2. Intel Scientific Computers. "iPSC/860 User's Guide." Order Number 311532-001, Beaverton, Oregon. June 1990.

3. Intel Scientific Computers. "Parallel Programming Primer." Order Number 311914-001, Beaverton, Oregon. March 1990.

4. Beam, R.M., and R.F. Warming. "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations." AIAA Journal, Vol. 16, No. 4. April 1978: 393-402.

5. MacCormack, R.W. "The Effect of Viscosity in Hypervelocity Impact Cratering." AIAA Paper, No. 69-354.

6. Opalka, K.O., and A. Mark. "The BRL-Q1D Code: A Tool for the Numerical Simulation of Flows in Shock Tubes with Variable Cross-Sectional Areas." Technical Report BRL-TR-2763. U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland. October 1986.

7. Hikida, S., R. Bell, and C. Needham. "The SHARC Codes: Documentation and Sample Problems." SSS-R-89-9878, S-CUBED, September 1988.

Intentionally Left Blank

No. of Copies	Organization
2	Administrator Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22304-6145
1	Commander U.S. Army Materiel Command ATTN: AMCAM 5001 Eisenhower Ave. Alexandria, VA 22333-0001
1	Director U.S. Army Research Laboratory ATTN: AMSRL-OP-SD-TA, Records Management 2800 Powder Mill Rd. Adelphi, MD 20783-1145
3	Director U.S. Army Research Laboratory ATTN: AMSRL-OP-SD-TL, Technical Library 2800 Powder Mill Rd. Adelphi, MD 20783-1145
1	Director U.S. Army Research Laboratory ATTN: AMSRL-OP-SD-TP, Technical Publishing Branch 2800 Powder Mill Rd. Adelphi, MD 20783-1145
2	Commander U.S. Army Armament Research, Development, and Engineering Center ATTN: SMCAR-TDC Picatinny Arsenal, NJ 07806-5000
1	Director Benet Weapons Laboratory U.S. Army Armament Research, Development, and Engineering Center ATTN: SMCAR-CCB-TL Watervliet, NY 12189-4050
1	Director U.S. Army Advanced Systems Research and Analysis Office (ATCOM) ATTN: AMSAT-R-NR, M/S 219-1 Ames Research Center Moffett Field, CA 94035-1000

No. of Copies	Organization
1	Commander U.S. Army Missile Command ATTN: AMSMI-RD-CS-R (DOC) Redstone Arsenal, AL 35898-5010
1	Commander U.S. Army Tank-Automotive Command ATTN: AMSTA-JSK (Armor Eng. Br.) Warren, MI 48397-5000
1	Director U.S. Army TRADOC Analysis Command ATTN: ATRC-WSR White Sands Missile Range, NM 88002-5502
1	Commandant U.S. Army Infantry School ATTN: ATSH-WCB-O Fort Benning, GA 31905-5000
	Aberdeen Proving Ground
2	Dir, USAMSAA ATTN: AMXSY-D AMXSY-MP, H. Cohen
1	Cdr, USATECOM ATTN: AMSTE-TC
1	Dir, USAERDEC ATTN: SCBRD-RT
1	Cdr, USACBDCOM ATTN: AMSCB-CII
1	Dir, USARL ATTN: AMSRL-SL-I
5	Dir, USARL ATTN: AMSRL-OP-AP-L

DISTRIBUTION LIST

No. of Copies	Organization
1	HQDA (SARD-TR/Ms. K. Kominos) Washington, DC 20310-0103
1	HQDA (SARD-TR/Dr. R. Chait) Washington, DC 20310-0103
1	Director of Defense Research & Engineering ATTN: DD/TWP Washington, DC 20301
1	Assistant Secretary of Defense (Atomic Energy) ATTN: Document Control Washington, DC 20301
1	Chairman Joint Chiefs of Staff ATTN: J-5, R&D Division Washington, DC 20301
2	Deputy Chief of Staff for Operations and Plans ATTN: Technical Library Director of Chemical and Nuclear Operations Department of the Army Washington, DC 20310
1	U.S. Army Research Development and Standardization Group (UK) ATTN: Dr. R. Reichenbach PSC 802, Box 15 FPO AE 09499-1500
1	Director Advanced Research Projects Agency ATTN: Technical Library 3701 North Fairfax Drive Arlington, VA 22203-1714
2	Director Federal Emergency Management Agency ATTN: Public Relations Office Technical Library Washington, DC 20472
1	Chairman DOD Explosives Safety Board Room 856-C Hoffman Bldg. I 2461 Eisenhower Avenue Alexandria, VA 22331-0600
1	Director Defense Intelligence Agency ATTN: DT-2/Wpns & Sys Div Washington, DC 20301
1	Director National Security Agency ATTN: R15, E. F. Butala Ft. George G. Meade, MD 20755
7	Director Defense Nuclear Agency ATTN: CSTI, Technical Library DDIR DFSP NANS OPNA SPSD SPTD Washington, DC 20305
3	Commander Field Command, DNA ATTN: FCPR FCTMOF FCTTS (Ed Martinez) Kirtland AFB, NM 87115
10	Central Intelligence Agency DIR/DB/Standard ATTN: GE-47 HQ (10 cps) Washington, DC 20505
2	Commander, USACECOM ATTN: AMSEL-RD AMSEL-RO-TPPO-P Fort Monmouth, NJ 07703-5301

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander, USACECOM R&D Technical Library ATTN: ASQNC-ELC-IS-L-R, Myer Center Fort Monmouth, NJ 07703-5000
1	Commander US Army Armament Research, Development and Engineering Center ATTN: SMCAR-FSM-W/Barber Bldg. 94 Picatinny Arsenal, NJ 07806-5000
1	Director US Army Missile and Space Intelligence Center ATTN: AIAMS-YDL Redstone Arsenal, AL 35898-5500
1	Commander National Ground Intelligence Center ATTN: Research & Data Branch 220 7th Street, NE Charlottesville, VA 22901-5396
1	Director US Army TRAC - Ft. Lee ATTN: ATRC-L, Mr. Cameron Fort Lee, VA 23801-6140
1	Director US Army Research Laboratory Materials Directorate ATTN: AMSRL-MA-B Watertown, MA 02172-0001
2	Commander US Army Strategic Defense Command ATTN: CSSD-H-MPL, Tech Lib CSSD-H-XM, Dr. Davies P.O. Box 1500 Huntsville, AL 35807
2	Commander US Army Natick Research and Development Center ATTN: AMDNA-D, Dr. D. Sieling STRNC-UE, J. Calligeros Natick, MA 01762
1	Commander US Army Engineer Division ATTN: HNDED-FD P.O. Box 1500 Huntsville, AL 35807
3	Commander US Army Corps of Engineers Waterways Experiment Station ATTN: CEWES-SS-R, J. Watt CEWES-SE-R, J. Ingram CEWES-TL, Tech Lib P.O. Box 631 Vicksburg, MS 39180-0631
1	Commander US Army Corps of Engineers Fort Worth District ATTN: CESWF-PM-J P.O. Box 17300 Fort Worth, Texas 76102-0300
1	Commander US Army Research Office ATTN: SLCRO-D P.O. Box 12211 Research Triangle Park, NC 27709-2211
3	Commander US Army Nuclear & Chemical Agency 7150 Heller Loop, Suite 101 Springfield, VA 22150-3198
1	Director HQ, TRAC RPD ATTN: ATRC-RPR, Radda Fort Monroe, VA 23651-5143
1	Director TRAC-WSMR ATTN: ATRC-WC, Kirby White Sands Missile Range, NM 88002-5502
1	Director TRAC-FLVN ATTN: ATRC Fort Leavenworth, KS 66027-5200

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander US Army Test & Evaluation Command ATTN: STEWS-NED (Dr. J L Meason) White Sands Missile Range, NM 88002-5158
2	Chief of Naval Operations Department of the Navy ATTN: OP-03EG OP-985F Washington, DC 20350
1	Commander Naval Electronic Systems Command ATTN: PME 117-21A Washington, DC 20360
1	Commander Naval Sea Systems Command ATTN: Code SEA-62R Department of the Navy Washington, DC 20362-5101
2	Office of Naval Research ATTN: Dr. A. Faulstick, Code 23 (2cps) 800 N. Quincy Street Arlington, VA 22217
1	Officer-in-Charge (Code L31) Civil Engineering Laboratory Naval Construction Battalion Center ATTN: Technical Library Port Hueneme, CA 93041
1	Commanding Officer (Code L51) Naval Civil Engineering Laboratory ATTN: J. Tancreto Port Hueneme, CA 93043-5003
1	Commander Dahlgreen Division Naval Surface Warfare Center ATTN: Code E23, Library Dahlgren, VA 22448-5000
1	Commander David Taylor Research Center ATTN: Code 522, Tech Info Ctr Bethesda, MD 20084-5000
1	Officer in Charge White Oak Warfare Center Detachment ATTN: Code E232, Technical Library 10901 New Hampshire Avenue Silver Spring, MD 20903-5000
1	Commanding Officer White Oak Warfare Center ATTN: Code WA501, NNPO Silver Spring, MD 20902-5000
1	Commander (Code 533) Naval Weapons Center ATTN: Technical Library China Lake, CA 93555-6001
1	Commander Naval Research Laboratory ATTN: Code 2027, Technical Library Washington, DC 20375
1	AL/LSCF ATTN: J. Levine Edwards AFB, CA 93523-5000
1	OLAC PL/TSTL ATTN: D. Shiplett Edwards AFB, CA 93523-5000
2	Air Force Armament Laboratory ATTN: AFATL/DOIL AFATL/DLYV Eglin AFB, FL 32542-5000
1	RADC (EMTLD/Docu Library) Griffiss AFB, NY 13441
3	Phillips Laboratory (AFWL) ATTN: NTE NTED NTES Kirtland AFB, NM 87117-6008
1	AFESC/RDCS ATTN: Paul Rosengren Tyndall AFB, FL 32403

DISTRIBUTION LIST

No. of Copies	Organization
1	AFIT ATTN: Technical Library, Bldg. 640/B Wright-Patterson AFB, OH 45433
1	AFIT/ENY ATTN: LTC Hasen, PhD Wright-Patterson AFB, OH 45433-6583
1	FTD/NIIS Wright-Patterson AFB, Ohio 45433
1	Director Idaho National Engineeri g Laboratory ATTN: Spec Programs, J. Patton 2151 North Blvd, MS 2802 Idaho Falls, ID 83415
2	Director Idaho National Engineering Laboratory EG&G Idaho Inc. ATTN: R. Guenzler, MS-3505 R. Holman, MS-3510 P.O. Box 1625 Idaho Falls, ID 83415
1	Director Lawrence Livermore National Laboratory ATTN: Dr. Allan Kuhl 2250 E. Imperial Highway, Suite # 650 El Segundo, CA 90245
1	Director Lawrence Livermore National Laboratory ATTN: Tech Info Dept L-3 P.O. Box 808 Livermore, CA 94550
2	Director Los Alamos National Laboratory ATTN: Th. Dowler, MS-F602 Doc Control for Reports Library P.O. Box 1663 Los Alamos, NM 87545
5	Director Sandia National Laboratories ATTN: Doc Control 3141 C. Cameron, Div 6215 A. Chabai, Div 7112 D. Gardner, Div 1421 J. McGlaun, Div 1541 P.O. Box 5800 Albuquerque, NM 87185-5800
1	Director Sandia National Laboratories Livermore Laboratory ATTN: Doc Control for Tech Library P.O. Box 969 Livermore, CA 94550
1	Director National Aeronautics and Space Administration ATTN: Scientific & Tech Info Fac P.O. Box 8757, BWI Airport Baltimore, MD 21240
1	Director NASA-Langley Research Center ATTN: Technical Library Hampton, VA 23665
1	Director NASA-Ames Research Center Applied Computational Aerodynamics Branch ATTN: Dr. T. Holtz, MS 202-14 Moffett Field, CA 94035
1	ADA Technologies, Inc. ATTN: James R. Butz Honeywell Center, Suite 110 304 Inverness Way South Englewood, CO 80112
1	Alliant Techsystems, Inc. ATTN: Roger A. Rausch (MN48-3700) 7225 Northland Drive Brooklyn Park, MN 55428

DISTRIBUTION LIST

No. of Copies	Organization
2	Applied Research Associates, Inc. ATTN: J. Keefer N.H. Ethridge P.O. Box 548 Aberdeen, MD 21001
1	Aerospace Corporation ATTN: Tech Info Services P.O. Box 92957 Los Angeles, CA 90009
3	Applied Research Associates, Inc. ATTN: R. L. Guice (3 cps) 7114 West Jefferson Ave., Suite 305 Lakewood, CO 80235
1	Black & Veatch, Engineers - Arcitects ATTN: H. D. Laverentz 1500 Meadow Lake Parkway Kansas City, MO 64114
1	The Boeing Company ATTN: Aerospace Library P.O. Box 3707 Seattle, WA 98124
1	California Research & Technology, Inc. ATTN: M. Rosenblatt 20943 Devonshire Street Chatsworth, CA 91311
1	Carpenter Research Corporation ATTN: H. Jerry Carpenter 27520 Hawthorne Blvd., Suite 263 P. O. Box 2490 Rolling Hills Estates, CA 90274
1	Dynamics Technology, Inc. ATTN: D. T. Hove G. P. Mason 21311 Hawthorne Blvd., Suite 300 Torrance, CA 90503
1	EATON Corporation Defense Valve & Actuator Div. ATTN: J. Wada 2338 Alaska Ave. El Segundo, CA 90245-4896
2	FMC Corporation Advanced Systems Center ATTN: J. Drotleff C. Krebs, MDP95 Box 58123 2890 De La Cruz Blvd. Santa Clara, CA 95052
1	Goodyear Aerospace Corporation ATTN: R. M. Brown, Bldg 1 Shelter Engineering Litchfield Park, AZ 85340
4	Kaman AviDyne ATTN: R. Ruetenik (2 cps) S. Criscione R. Milligan 83 Second Avenue Northwest Industrial Park Burlington, MA 01830
3	Kaman Sciences Corporation ATTN: Library P. A. Ellis F. H. Shelton P.O. Box 7463 Colorado Springs, CO 80933-7463
2	Kaman-Sciences Corporation ATTN: DASIAC (2cps) P.O. Drawer 1479 816 State Street Santa Barbara, CA 93102-1479
1	Ktech Corporation ATTN: Dr. E. Gaffney 901 Pennsylvania Ave., N.E. Albuquerque, NM 87111
1	Lockheed Missiles & Space Co. ATTN: J. J. Murphy, Dept. 81-11, Bldg. 154 P.O. Box 504 Sunnyvale, CA 94086

DISTRIBUTION LIST

No. of Copies	Organization
2	McDonnell Douglas Astronautics Corporation ATTN: Robert W. Halprin K.A. Heinly 5301 Bolsa Avenue Huntington Beach, CA 92647
1	MDA Engineering, Inc. ATTN: Dr. Dale Anderson 500 East Border Street Suite 401 Arlington, TX 76010
1	Orlando Technology, Inc. ATTN: D. Matuska 60 Second Street, Bldg. 5 Shalimar, FL 32579
2	Physics International Corporation P.O.Box 5010 San Leandro, CA 94577-0599
1	R&D Associates ATTN: G.P. Ganong P.O. Box 9377 Albuquerque, NM 87119
1	Science Applications International Corporation ATTN: J. Guest 2301 Yale Blvd. SE, Suite E Albuquerque, NM 87106
1	Science Applications International Corporation ATTN: N. Sinha 501 Office Center Drive, Apt. 420 Ft. Washington, PA 19034-3211
2	Science Center Rockwell International Corporation ATTN: Dr. S. Chakravarthy Dr. D. Ota 1049 Camino Dos Rios P.O. Box 1085 Thousand Oaks, CA 91358
2	S-CUBED A Division of Maxwell Laboratories, Inc. ATTN: C. E. Needham L. Kennedy 2501 Yale Blvd. SE Albuquerque, NM 87106
3	S-CUBED A Division of Maxwell Laboratories, Inc. ATTN: Technical Library R. Duff K. Pyatt PO Box 1620 La Jolla, CA 92037-1620
1	Sparta, Inc. Los Angeles Operations ATTN: I. B. Osofsky 3440 Carson Street Torrance, CA 90503
1	Sunburst Recovery, Inc. ATTN: Dr. C. Young P.O. Box 2129 Steamboat Springs, CO 80477
1	Sverdrup Technology, Inc. ATTN: R. F. Starr P. O. Box 884 Tullahoma, TN 37388
1	Sverdrup Technology, Inc. Sverdrup Corporation-AEDC ATTN: B. D. Heikkinen MS-900 Arnold Air Force Base, TN 37389-9998
3	SRI International ATTN: Dr. G. R. Abrahamson Dr. J. Gran Dr. B. Holmes 333 Ravenswood Avenue Menlo Park, CA 94025
1	Texas Engineering Experiment Station ATTN: Dr. D. Anderson 301 Engineering Research Center College Station, TX 77843

DISTRIBUTION LIST

No. of Copies — Organization

1 Thermal Science, Inc.
ATTN: R. Feldman
2200 Cassens Dr.
St. Louis, MO 63026

2 Thinking Machines Corporation
ATTN: G. Sabot
R. Ferrel
245 First Street
Cambridge, MA 02142-1264

1 TRW
Ballistic Missile Division
ATTN: H. Korman,
Mail Station 526/614
P.O. Box 1310
San Bernadino, CA 92402

1 Battelle
TWSTIAC
505 King Avenue
Columbus, OH 43201-2693

1 California Institute of Technology
ATTN: T. J. Ahrens
1201 E. California Blvd.
Pasadena, CA 91109

2 Denver Research Institute
ATTN: J. Wisotski
Technical Library
P.O. Box 10758
Denver, CO 80210

1 Massachusetts Institute of Technology
ATTN: Technical Library
Cambridge, MA 02139

2 University of Maryland
Institute for Advanced Computer Studies
ATTN: L. Davis
G. Sobieski
College Park, MD 20742

No. of Copies — Organization

1 University of Minnesota
Army High Performance Computing Research Center
ATTN: Dr. Tayfun E. Tezduyar
1100 Washington Ave. South
Minneapolis, Minnesota 55415

3 Southwest Research Institute
ATTN: Dr. C. Anderson
S. Mullin
A. B. Wenzel
P.O. Drawer 28255
San Antonio, TX 78228-0255

1 Stanford University
ATTN: Dr. D. Bershader
Durand Laboratory
Stanford, CA 94305

1 State University of New York
Mechanical & Aerospace Engineering
ATTN: Dr. Peyman Givi
Buffalo, NY 14260

Aberdeen Proving Ground

1 Cdr, USATECOM
ATTN: AMSTE-TE-F, L. Teletski

1 Cdr, USATHMA
ATTN: AMXTH-TE

1 Cdr, USACSTA
ATTN: STECS-LI

DISTRIBUTION LIST

26 Dir,USARL
ATTN: AMSRL-CI-A, H. Breaux
AMSRL-CI-AC, R. Sheroke
AMSRL-CI-AD,
C. Nietubicz
C. Ellis
AMSRL-CI-C, W. Sturek
AMSRL-CI-CA,
M. Coleman
N. Patel
AMSRL-CI-S, R. Pearson
AMSRL-CI-SA, T. Kendall
AMSRL-WT, D. Hisley
AMSRL-WT-N, J. Ingram
AMSRL-WT-NA, A. Kehs
AMSRL-WT-NB, J. Gwaltney
AMSRL-WT-NC,
R. Lottero
B. McGuire
A. Mihalcin
K. Opalka
R. Raley
M. Unekis
AMSRL-WT-ND, J. Miletta
AMSRL-WT-NF, L. Jasper
AMSRL-WT-NG, T. Oldham
AMSRL-WT-NH, J. Corrigan
AMSRL-WT-PB,
P. Weihnacht
B. Guidos
AMSRL-WT-TC, K. Kimsey

INTENTIONALLY LEFT BLANK.

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number ARL-TR-589 Date of Report October 1994

2. Date Report Received _____

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) _____

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) _____

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. _____

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) _____

CURRENT ADDRESS

_____
Organization

_____
Name

_____
Street or P.O. Box No.

_____
City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD ADDRESS

_____
Organization

_____
Name

_____
Street or P.O. Box No.

_____
City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)
(DO NOT STAPLE)